

SCR.2.1 TP 16 ⊥ :

IMUNES

Un émulateur/simulateur de topologies de réseaux

http ://imunes.net/

IMUNES constitue un cadre pour émuler des nœuds virtuels. Ces nœuds virtuels peuvent être interconnectés dans le noyau de la machine hôte pour simuler une topologie de réseau.

Se connecter maintenant sur le compte local et y créer le répertoire `~/IMUNES/TP16`. C'est dans ce répertoire que seront enregistrés les fichiers de la séance courante de TP.

Si on n'est pas sur sa vm attitrée, alors réaliser la procédure suivante à la fin de la séance :

DEBUT.

1. Se connecter sur son compte personnel pour y créer le répertoire `~/SCR/` s'il n'existe pas déjà.
2. Se connecter sur le compte local et transférer le répertoire `IMUNES/TP16` vers son compte personnel (répertoire `~/SCR/`) par `sftp`, `put -r TP16`, puis supprimer le contenu du répertoire `IMUNES` du compte local.

FIN.

I. Démarrage.

Selon l'installation actuelle des salles machines, l'utilisation de IMUNES se fait en étant sur le compte local et en lançant : `sudo imunes&`

1. Lancer donc `imunes` et construire la première topologie de réseau fournie en fin d'énoncé en utilisant les mêmes noms et adresses. Sauvegarder sous le nom `simple1.imn`
2. Lancer le réseau virtuel à l'aide de *Experiment* -> *Execute*

Sur la machine host, on peut alors voir l'identifiant de l'instance lancée :

```
$ sudo himage -l
i6fd73 (simple1.imn - Fri May 17 15:47:35 CEST 2019)
```

3. Pour lancer des commandes sur un nœud virtuel, disons la commande `ls` sur `pc1`, on peut faire (1) bouton droit de la souris -> *shell window* pour avoir un terminal sur `pc1` ou alors (2) depuis la ligne de commande de la machine host :

```
$ sudo himage pc1@i6fd73 ls
bin  boot.conf  etc  lib      media  opt      proc  run  srv  tmp  var
boot dev  home  lib64  mnt     out.log  root  sbin sys  usr
```

Note. S'il n'y a qu'une seule instance de topologie virtuelle contenant le nom `pc1`, il n'est pas nécessaire d'ajouter `@...` après le nom `pc1`.

4. Sans ouvrir de terminaux sur les nœuds virtuels, lancer `tcpdump` sur `pc1` et sur `pc3` (bouton droit de la souris), puis à la ligne de commande de la machine host, lancer un ping de `pc1` vers `pc3`. Les traces montrées par `tcpdump` sont-elles cohérentes ?
5. On peut terminer une execution soit (1) par *Experiment* -> *Terminate* si l'instance est celle contrôlée par l'interface graphique, soit (2) à la ligne de commande de la machine host :

```
$ sudo imunes -b -e i6fd73
```

II. Avec une passerelle.

1. Réaliser la deuxième topologie de réseau fournie en fin d'énoncé en utilisant les mêmes noms et adresses. Sauvegarder sous le nom `simple2.imn`
2. À l'aide de l'onglet *configuration* et **sans utiliser de route par défaut**, configurer l'ensemble pour que les machines du segment S1 : 172.16.1.0/24 puissent initier des communications vers les machines du segment S2 : 172.16.2.0/24, qui doivent alors y répondre, et vice-versa.

On teste si tout est bien configuré.

III. Avec deux passerelles.

Le réseau d'une organisation a la configuration suivante :

- Une passerelle P1 fait le lien entre deux segments S1 : 192.168.0.0/18 et S2 : 192.168.64.0/18.
 - Une passerelle P2 fait le lien de S2 avec S3 : 192.168.128.0/18 et S4 : 192.168.192.0/18
 - Les communications doivent aboutir quelque soient la source et la destination.
1. Créer le fichier `simple3.imn` correspondant en plaçant deux pc sur chacun des segments S1, S2, S3 et S4. Donner des adresses cohérentes à l'ensemble des interfaces.
 2. À l'aide de l'onglet *configuration*, configurer l'ensemble en plaçant un nombre minimal d'entrées dans chaque table de routage.

Tester l'ensemble.

Avant de quitter la salle, vérifier si on est concerné par la procédure indiquée en début d'énoncé afin de la réaliser si besoin.

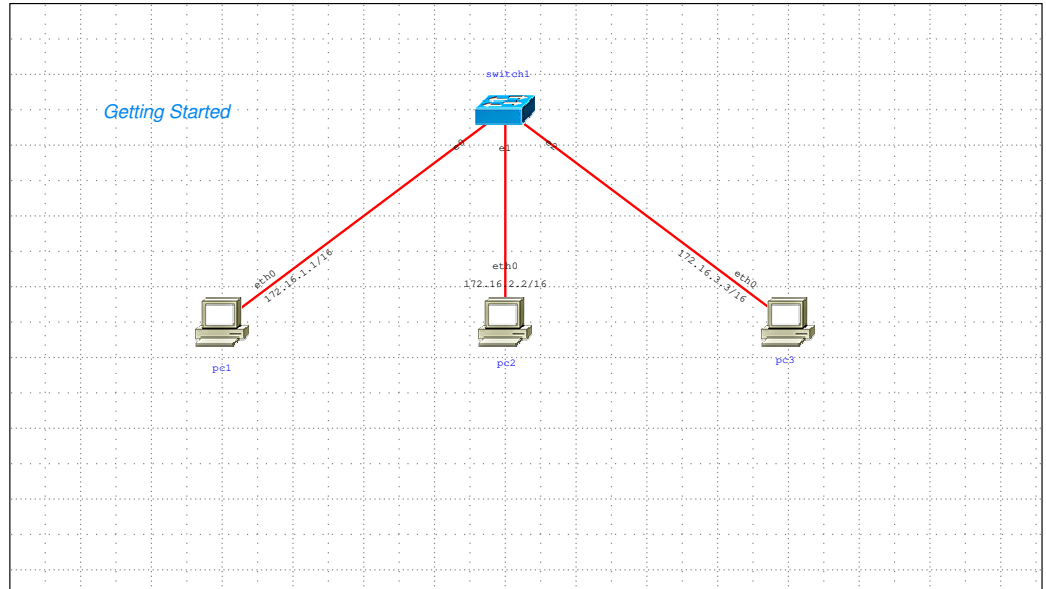


FIGURE 1 – above : simple1 - below : simple2

